

# One pixel attack for fooling deep neural networks

Jiawei Su  
Kyushu University  
Japan

jiawei.su@inf.kyushu-u.ac.jp

Danilo Vasconcellos Vargas  
Kyushu University  
Japan

vargas@inf.kyushu-u.ac.jp

Sakurai Kouichi  
Kyushu University  
Japan

sakurai@csce.kyushu-u.ac.jp

## Abstract

Recent research has revealed that the output of Deep neural networks(DNN) is not continuous and very sensitive to tiny perturbation on the input vectors and accordingly several methods have been proposed for crafting effective perturbation against the networks. In this paper, we propose a novel method for optically calculating extremely small adversarial perturbation (few-pixels attack), based on differential evolution. It requires much less adversarial information and works with a broader classes of DNN models. The results show that 73.8% of the test images can be crafted to adversarial images with modification just on one pixel with 98.7% confidence on average. In addition, it is known that investigating the robustness problem of DNN can bring critical clues for understanding the geometrical features of the DNN decision map in high dimensional input space. The results of conducting few-pixels attack contribute quantitative measurements and analysis to the geometrical understanding from a different perspective compared to previous works.

## 1. Introduction

In the domain of image recognition, DNN-based approach has overcome traditional image processing techniques and achieved even human-competitive results [9]. However, several studies have revealed that artificial perturbations on natural images can easily make DNN misclassify and accordingly proposed effective algorithms for generating such samples called “adversarial images” [1, 2, 3, 4]. A main way of creating adversarial images is adding a tiny amount of well-tuned additive perturbation to a correctly classified natural image that is expected to be imperceptible to human eyes. Such modification can cause the classifier to label the modified image as completely something else. However, most of previous attacks did not consider very limited adversarial cases, namely the amount of modifications are sometimes perceptible to human eyes in practice (see fig.2 for an example). In addition, investigating

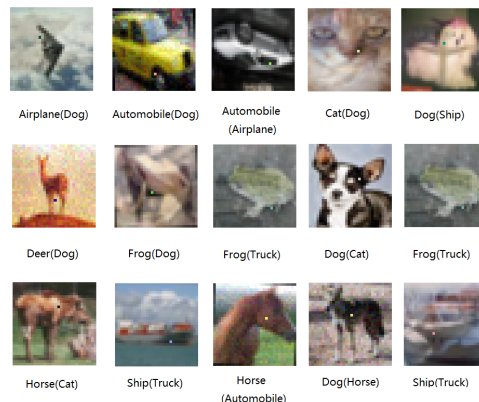


Figure 1. Adversarial images generated by our algorithm that successfully fooled the target DNN with one pixel modification. The labels inside the brackets are the target classes and outside are their original classes. The pixels modified might not be obvious so readers have to look carefully.

adversarial images created under limited scenario is more interesting since they are likely closer to the boundaries between source and target classes and investigating such critical points might give new insight about the geometrical features of DNN input space [23].

In this paper, by perturbing few pixels (1, 3 or 5 pixels out of 1024) with differential evolution, we propose a black-box DNN attack, which we call it “few-pixels attack”, in a scenario where the only information available are the probability labels. Our proposal mainly has the following advantages compared to previous works:

- Being able to launch non-targeted attacks by only modifying 1, 3 and 5 pixels, with the success rates of 73.8%, 82.0% and 87.3% respectively and 98.7% probability label of target classes on average.
- Requiring only black-box feedback (probability labels) but no inner information of target DNN such as gradients and network structure. Our method is also simpler since it does not abstract the problem of

searching perturbation to any explicit target functions to solve but directly focus on improving the probability label values of the target classes.

- Can attack a broader classes of DNNs (e.g. networks that are not differentiable or when the gradient calculation is difficult).

There are two main reasons that we consider few-pixels attack. 1) Few-pixel attack can be effectiveness for hiding the modification in practice. To the best of our knowledge, none of the previous works can guarantee that the perturbation made can be completely invisible. A direct way of mitigating this problem is to limit the perturbation as smaller as possible. Specifically, instead of theoretically proposing additional constraints or considering more complex cost functions of perturbation, we propose an empirical solution by seriously controlling the number of pixels that can be modified, in specific 1, 3 and 5 pixels out of an 32 X 32 image, namely we use the number of pixels as units instead of length of perturbation vector to measure the perturbation strength and consider the worst case which is one-pixel modification as well as two other scenarios (i.e. 3 and 5 pixels) for comparison. 2) Geometrically, several previous works have analyzed the vicinity of natural images by limiting the strength of total pixel modifications. For example, the universal perturbation adds small values to each pixel such that it searches the adversarial images in a sphere region around the natural image [24]. On the other side, the proposed few-pixel perturbations can be regarded as cutting the input space using very low-dimensional slices, which is a different way of exploring the features of DNN input space.

According to the experimental results, the main contributions of our work include:

- **The effectiveness of conducting non-target attack using few-pixel attack.** We show that with only 1 pixel modification, there are 73.8% of the images can be perturbed to one or more target classes, 82.0% and 87.3% in the cases of 3 and 5-pixel attacks. We show the non-sensitive images are even much rarer than sensitive images even if limiting the perturbation to such a small scope, therefore few-pixel modification is an effective method of searching adversarial images while can be hardly recognized by human eyes in practice.
- **The number of target classes that a natural image can camouflage.** In the case of 1 pixel perturbation, each natural image can be perturbed to 2.3 other classes on average. In specific, there are 18.4%, 17.2% and 16.6% of the images can be perturbed to 1, 2, 3 target classes. In the case of 5-pixel perturbation, the amounts of images that can be perturbed to from 1 to 9 target classes become almost even.

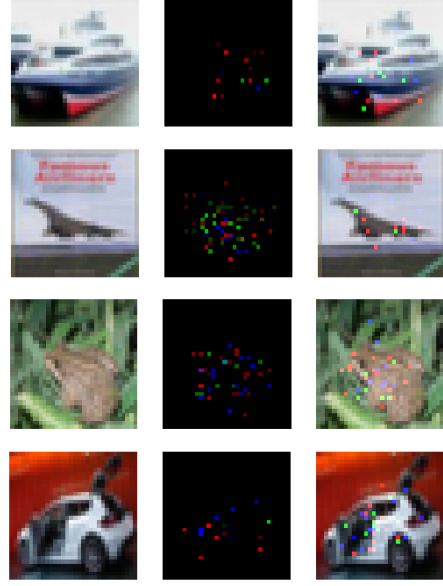


Figure 2. An illustration of the adversarial images generated by using[1]. The perturbation is conducted on about 4% of the total pixels and can be easily recognized by human eyes. Since the adversarial pixel perturbation has become a common way of generating adversarial images, such abnormal “noise” can be recognized with expertise.

- **Similar perturbation direction to a specific target class.** Effectiveness of universal perturbation has shown that many images can be perturbed through similar directions such that decision boundaries might leak diversity [24] while our results show that data-points belonging to same classes can be always more easily perturbed to specific classes with same amount of perturbations (i.e. 1, 3 or 5 pixel-modifications).
- **Geometrical understanding of data-point distribution in high dimensional input space.** Geometrically, the information obtained by conducting few-pixel attack can be also regarded as a quantitative result on changes in class labels on the cross sections obtained by using simply low dimensional slices to cut the input space. In particular, our results indicate that some decision domains might have very great depths towards many different directions but inside these deep areas, the decision domains are quite narrow. In other words, these domains can have many long and thin extended synapses towards different directions in the input space.

In the following, we abbreviate the originally true class of an adversarial image as “original class”, the objective class that adversaries desire the DNN to recognize their ad-

versarial images as “target class”, the DNN classifier that adversaries want to fool as “target system”. Mainly in Section.4, for a specific class  $C$ , we sometimes call perturbing a data-point to another target class as “going out” from  $C$  and perturbing a data-point of another class to  $C$  as “getting into”  $C$ .

## 2. Related works

The security problem of DNN has become a critical topic [6] [14]. C. Szegedy et al. first reveal the weakness of sensitivity to small well-turned artificial perturbation [3], which can be crafted by several gradient-based algorithms relying on reusing the back-propagation procedure for obtaining gradient information [2, 3]. Specifically, I.J. Goodfellow et al. proposed “fast gradient sign” algorithm for calculating effective perturbation based on a hypothesis that the linearity and high-dimensions of inputs are the main reason that a broad class of networks are sensitive to small perturbation [2]. S.M. Moosavi-Dezfooli et al. proposed a greedy perturbation searching method by assuming the linearity of the DNN decision boundaries [4]. In addition, N. Papernot et al. utilize Jacobian matrix to build “Adversarial Saliency Map” which indicates the effectiveness of conducting a fixed length perturbation through the direction of each axis [1, 18]. Another kind of adversarial image is also proposed by A. Nguyen et al. [7]. The images can hardly be recognized by human eyes but nevertheless classified by the network with high confidences.

Several black-box attacks that require no knowledge about the target systems such as gradients have also been proposed [?][21][13]. In particular, to the best of our knowledge, the only work that ever mentioned 1-pixel attack before us is carried out by N. Narodytska et al [13]. However, being different from our work, they only utilized it as a starting point to derive a further black box attack which needs to modify more pixels (i.e. about 30 pixels), but did not formally consider the scenario of 1-pixel attack and conducted the experiments. In particular their attack with 1-pixel is quite coarse which is almost equivalent to a random search. In addition, they did not systematically measure the effectiveness of the attack and obtain quantitative results for evaluation. They also did not derive any geometrical features from 1-pixel attack and accordingly make further discussion.

Many efforts have been paid for understanding DNN by visualizing the activation of network nodes [15, 16, 17], on the otherside the geometrical characteristics of DNN boundary have gained less attraction due to the difficulty of understanding high-dimensional space. However the robustness evaluation of DNN with respect to adversarial perturbation might shed light to this complex problem [23]. For example, both natural and random images are found to be vulnerable to adversarial perturbation. Assume these images are

evenly distributed, it suggests that most data-points in the input space are gathered near to the boundaries and deep inside the domain is quite hollow [23]. In addition, A. Fawzi et al. reveal more clues are obtained by curvature analysis including the connectivity of regions of same classes, the region among most directions around natural images are flat, with few directions that the space is curved where the images are sensitive to perturbation [22]. On the otherside, universal perturbation (i.e. a perturbation universally effective for generating adversarial images from different natural images) is proved to be effective while random perturbation is comparatively much less effective, which indicates the diversity of boundaries might be low and the patterns of shapes of boundaries near to different data points are similar [24].

## 3. Methodology

### 3.1. Problem description

Generating adversarial images can be formalized as a simple optimization problem with constraints. We assume an input image can be interpreted by a vector which each scalar element represents one pixel. Let  $f$  be the target image classifier that receives  $n$ -dimensional inputs,  $\mathbf{x} = (x_1, \dots, x_n)$  be the original natural image correctly classified as class  $t$  and vector  $\mathbf{e}(\mathbf{x}) = (e_1, \dots, e_n)$  is an additive perturbation according to  $\mathbf{x}$ . The classifier  $f$  is a mapping that separates the input space into classification regions. The goal of adversaries in the case of target attack is to find the optimized  $e(x)$  for the following question.

$$\begin{aligned} & \underset{\mathbf{e}(\mathbf{x})}{\text{minimize}} \quad \|\mathbf{e}(\mathbf{x})\| \\ & \text{subject to} \quad f(\mathbf{x} + \mathbf{e}(\mathbf{x})) \neq f(\mathbf{x}) \end{aligned}$$

For obtaining the optimized perturbation vector  $\mathbf{e}(\mathbf{x})$ , one needs to decide how many dimensions and which dimensions that need to perturb and the corresponding strength of modification. Many kinds of perturbations proposed only modify a part of total dimensions such that a considerable number of elements of  $\mathbf{e}(\mathbf{x})$  are left to zeros. In our case, the numbers of dimensions that need to perturb are set to be constant numbers 1, 3 and 5, and the proposed method is utilized for solving the other two variables. We do not set constraints on the strength of perturbation on each dimension.

Geometrically, the entire input space of a DNN is a high-dimensional cube. The 1-pixel modification proposed can be seen as perturbing the data-point towards the parallel direction to the axis of one of the  $n$  dimensions. Similarly, the 3(5)-pixel modification moves the data-points within 3(5)-dimensional cubes. Overall, such perturbations with regard to few pixels are conducted on the low-dimensional slices of input space. Intuitively, it seems that 1-pixel perturba-

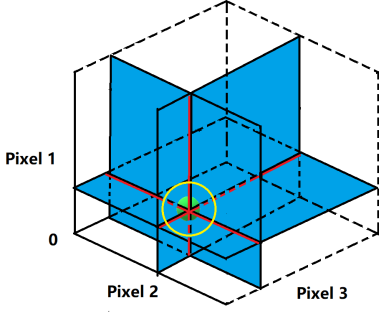


Figure 3. An illustration of using 1 and 2-pixel perturbation attack in a 3-dimensional input space (i.e. the image has three pixels.) The green point denotes a natural image to perturb. In the case of 1-pixel perturbation, the search space is on the three lines intersected at the natural image and each of three lines is perpendicular to any others, denoted by red and black stripes. In the case of 2-pixel, the search space is on three 2-dimensional planes intersected and each of three planes is perpendicular to any others, denoted by blue. To sum it up, 1 and 2-pixel attack search the perturbation on 1 and 2 dimensional slices of the original 3-d input space. In addition, the yellow circle indicates the search areas of previous works which utilized  $L_p$  norm to control the overall modification space. Comparatively, our few-pixel attack can search much further areas.

tion is merely a small change in the vast high dimensional input space such that it might hardly be able to find the effective perturbations. In fact 1-pixel perturbation allows to modify a image towards a chosen direction out of  $n$  possible directions that parallel to the axis of input space coordinate, with arbitrary strength. Therefore the complete set of all adversarial images that can be created from an image, forms a  $n$ -dimensional coordinate inside the input space which the origin is the image itself and each axis has the same length to the input space coordinate. In other words the search of adversarial images starts from the original image and go across the input space through the directions which each of them is perpendicular to any others. This is illustrated by fig.3 in the case when  $n = 3$ . Therefore, even if only modifying one pixel, it allows to search the candidate adversarial images in a fairly wide scope. Compared with previous works using  $L_p$  norm to control the overall perturbation strength, which the resulting search spaces are a small sphere around the vicinity of the image, the search of few-pixel attack can go much further in the input space therefore more hopeful of reaching other targeted classes.

As an extreme case, the university noise perturbation directly operates in the original input space by modifying all pixels with an overall constraint on the strength of accumulated modification [24, 25]. The few-pixel attack considered

in the research is the opposite which specifically focus on few pixels but does not limit the strength of modification. Geometrically, the universal perturbation moves small steps in the original input space, while few-pixel attacks use very low dimensional slices to cut the input space and evaluate how class label changes on the resulting cross sections.

### 3.2. Genetic algorithm and differential evolution

Differential evolution (DE) which belongs to the general class of genetic algorithm (GA), is a solution for solving optimized problems by keeping improving the quality of a candidate population according to a given fitness function [12]. Even if DE does not guarantee a globally optimal solution is ever found, it has mechanisms in the population selection phase that keep the diversity such that in practice it is expected to work better on finding higher quality solutions than gradient-based solutions, or even other kinds of GAs. In specific, during each iteration another group of new population (children) is generated according to the population generated from last iteration (fathers). Then the children are compared with their corresponding fathers, survive if they are better than their fathers. In such a way of only comparing the father and his child, the goal of keeping diversity and improving fitness values can be simultaneously achieved.

DE does not use the gradient information for optimizing therefore does not require for the optimization problem to be differentiable such that it can be utilized on a wider range of optimization problems compared to gradient based methods such as those are not continuous, time-related, noisy and so on. Using DE for generating adversarial images can generally have the following advantages:

- **Higher probability of finding global optima.** Since we put strict constraint on number of pixels that can be perturbed, this requires finding adversarial perturbation with better quality than previous. Prior solutions based on gradient descent or greedy search who can get stuck in local optima might not satisfy. On the other side, differential evolution is known for finding global optima due to inner mechanisms to keep the variety.
- **Require less information from target system.** DE does not require for the optimization problem to be differentiable as is required by classical optimization methods such as gradient descent and quasi-newton methods. This is critical in the case of generating adversarial images since 1) There are networks that are not differentiable, for instance [5]. 2) Calculating gradient requires much more capability of access to the victim systems, which can be hardly realistic in many cases. Although training an approximated model

through the black-box reaction from the targets is theoretically possible, the accuracy of approximation can be influenced by many factors in practice while time and resource consuming. In the other side, using DE requires neither gradient information nor training any additional models. It can directly craft adversarial images based on only the black-box reaction from the victim.

- **Simplicity.** Most of previous works abstract the problem of searching the effective perturbation to specific optimization problem (e.g. an explicit target function with constraints). Namely they made additional assumptions to the searching problem and might bring additional complexity. Our method does not solve any explicit target functions but directly works with the probability label value of the target classes.

### 3.3. Settings

We encode the information of perturbation into DNAs as input to be evolved by differential evolution. One DNA contains a fixed number of perturbation and each perturbation is a tuple holding 5 elements: x-y coordinates and RGB value of the perturbation. One perturbation modifies one pixel. The initial number of DNAs (population) is 400 and at each iteration another 400 children will be produced by using the following formula:

$$x_i(g+1) = x_{r1}(g) + F(x_{r2}(g) + x_{r3}(g)), \quad r1 \neq r2 \neq r3$$

The  $x_i$  is an element of a child,  $r1, r2, r3$  are random numbers,  $F$  is the scale parameter set to be 0.5,  $g$  is the current index of generation. Once generated, the children compete with their corresponding fathers according to the index of the population and the winners survive until next iteration. The maximum number of iteration is set to be 75 and an early-stop criteria will be triggered when the probability label of target class exceeded 99%. The initial population is initialized by using uniform distributions  $U(1, 32)$  for generating x-y coordinate (i.e. the image is with a size of 32X32) and Gaussian distributions  $N(128, 75)$  for RGB values. Since we aim to launch the targeted attack, the fitness function is simply the probabilistic label of the target class.

## 4. Evaluation and results

We consider both the scenarios of targeted and non-targeted attacks. We trained a target DNN image recognizer by using cifar10 data set [11], with the parameter setting listed in Table.1. The trained DNN has 86.7% test accuracy. The trained model resembles the model proposed by J.T.Springenberg et al. [19] which gets rid of max-pooling layers to train simpler but more effective DNNs.

```
conv2d layer(kernel=3, stride = 1, depth=96)
conv2d layer(kernel=3, stride = 1, depth=96)
conv2d layer(kernel=3, stride = 2, depth=96)
conv2d layer(kernel=3, stride = 1, depth=192)
conv2d layer(kernel=3, stride = 1, depth=192)
    dropout(0.3)
conv2d layer(kernel=3, stride = 2, depth=192)
conv2d layer(kernel=3, stride = 2, depth=192)
conv2d layer(kernel=1, stride = 1, depth=192)
conv2d layer(kernel=1, stride = 1, depth=10)
    average pooling layer(kernel=6, stride=1)
        flatten layer
            softmax classifier
```

Table 1. The structure of the target network

We utilize DE to evolve images to generate adversarial images utilized for both targeted and non-targeted attacks. 1000 candidate natural images that involve the evolution are randomly selected from the test data set of cifar-10 [11]. For each image, we try to perturb it to other 9 target class respectively. We implement three kinds of perturbation to modify the image: 1, 3 and 5 pixels. This leads to the total of 27000 adversarial images created. We introduce several kinds of measures for evaluating the effectiveness of the attack:

- **Success rate.** In the case of targeted(non-targeted) attack, it is defined as the percentage of adversarial images that were successfully classified by the DNN as the specific(arbitrary) target class.
- **Adversarial probability labels** accumulates the values of probability label of the target class for each successful perturbation, then divided by the total number of adversarial images generated.
- **Number of target classes** counts the number of natural images that successfully perturb to a certain number (i.e. from 0 to 9) of target classes. In particular, by counting the number of images that can not be perturbed to any other classes, the effectiveness of non-targeted attack can be evaluated.
- **Number of original-target class pairs** counts the number of times of each original-destination class pairs being fooled.

### 4.1. Results

The success rates and adversarial probability labels for 1, 3 and 5-pixel perturbations are shown by table.2. The number of target classes is shown by fig.4. The number of original-target class pairs is shown by the heat-maps of fig.6. In addition to the number of original-target class pairs,

	1 pixel	3 pixels	5 pixels
Success rate(tar)	23.46%	35.52%	42.78%
Success rate(non-tar)	73.8%	82.0%	87.3%
Accumulated labels	23.08%	35.08%	42.36%
Rate/Labels	98.37%	98.76%	99.02%

Table 2. Measures for evaluating the effectiveness of generating adversarial perturbation.

the total time of each class involving successful perturbations as the initial/target class is shown by fig.7.

#### 4.1.1 Success rate and adversarial probability labels (Targeted attack results)

The success rate shows that the perturbation can cause a considerable amount of successful crafting of adversarial images. In specific, on average each natural image can be perturbed to at least 2, 3 and 4 target classes by 1, 3 and 5 pixel modification. By dividing the adversarial probability labels by the success rates, it can be aware that for each successful perturbation, our algorithm averagely provides 98.7% probability label to the target classes.

#### 4.1.2 Number of target classes(Non-targeted attack results)

Regarding the results shown by fig.4 for number of target classes, we find that even after 1-pixel modification, a fair amount of natural images can be perturbed to 2, 3 and 4 target classes. By increasing the number of pixels modified, perturbation to more target classes becomes possible. For launching non-targeted attack, the success rates of perturbing an arbitrary image to at least one target classes are 73.8%, 82.0% and 87.3% in three cases respectively. Such results are competitive with previous non-targeted attack methods which need much more distortions than ours. Geometrically, it shows that using 1, 3 and 5-dimensional slices to cut the input space is enough to find the corresponding adversarial images for most of natural images.

The appearances of data-points that can be simultaneously perturbed to multiple classes show the effectiveness of searching adversarial images on very low-dimensional slices. In the case of 1-pixel modification, natural images can be commonly perturbed to 1, 2 and 3 target classes with almost even probability. By increasing the number of pixels up to 5, a considerable number of images can be even simultaneously perturbed to 8 target classes. In some rare cases, an image can go to all target classes with 1-pixel modification (see Fig.8.)

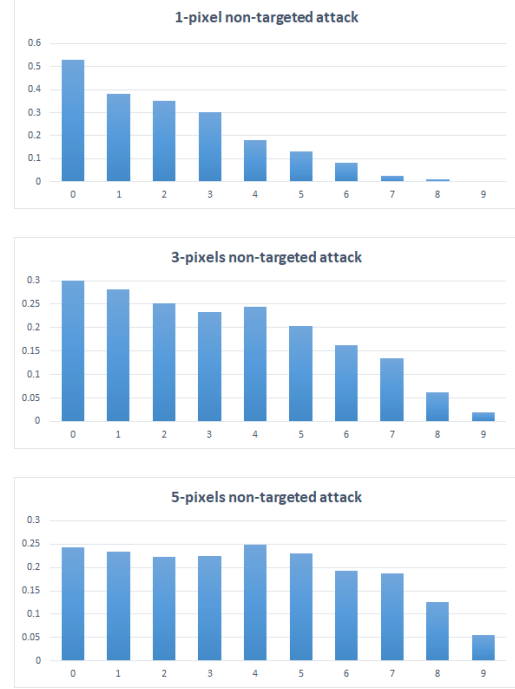


Figure 4. The graphs count the number of images that successfully perturbed to a certain number (from 0 to 9) of target classes by using 1, 3 and 5-pixel perturbation respectively, divided by number of images for normalization, shown by the vertical axis. The horizontal axis indicates the number of target classes.

#### 4.1.3 Original-target class pairs

From fig.6 it can be seen that there are specific original-target class pairs that are much more vulnerable than others, for example the images of cat (class 3) can be much more easily perturbed to dog (class 5), but can hardly reach the automobile (class 1). This indicates that the vulnerable target classes (directions) are shared by different data-points that belong to the same class therefore possibly similar shapes of boundaries in the vicinity of data-points along the directions to different target classes. A similar conclusion has been made by S. M. Moosavi-Dezfooli et al. [25] by proving that the boundaries lack diversity by showing the ineffectiveness of conducting perturbations to random directions. In specific, if the shapes of decision boundaries are diverse and irregular near different data-points, the random perturbation should be the best strategy and obtain much higher success rate of attack.

On the other side, in the case of 1-pixel attack, some classes are more robust than others since their data-points can be relatively hard to perturb to other classes. Among these data-points, there are points that can not be perturbed to any other classes. This indicates that the labels of these points can not be changed when going across the input space through  $n$  directions therefore the depths of the corre-



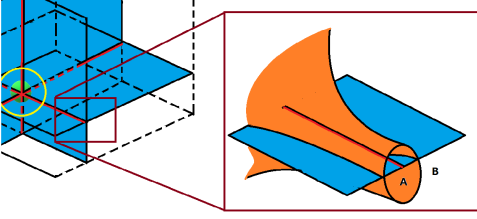


Figure 5. This illustration is an extension to fig.3. We conjecture that some decision domains have many long extension areas with a shape of a cylinder towards different directions according to the existence of robust data-points that can not be perturbed to any other classes in the case of 1-pixel attack. In the figure, the orange pipe-shape boundary is actually the decision boundary between class A (inside the pipe) and B. These pipe-shape boundaries are probably thin and do not have enough widths towards other dimensions. This is proved by that fact that when simply increasing the dimension of the attack, the amount of such robust data-points decreases.

sponding decision domain along these directions are great. However, these deep areas might not have enough width towards other directions since by increasing the number of pixels, the amount of robust points reduced. We therefore conjecture that the decision domain of these data-points might have many thin synapses to different directions. This is illustrated by fig.5.

It can be seen that each heat-map matrix of fig.6 is approximately symmetric, indicates that each class averagely has similar times of in and out, but we also see an obvious exception: the cat class (class 3). The class can be easily gone out but relatively hardly be got into. Such unbalance is intriguing since it indicates the cat is similar to most of the other classes like frog and dog but not vice-versa. This might due to the data-points of cat class are closer to the boundaries than others. In practice, such vague classes are more vulnerable since they are ideal start points of perturbation to many other classes.

## 5. Discussion and future work

We discuss some further correlations with related works. Previous results have shown that many data-points might be located near to the boundaries [23] through the change in class labels by conducting a small move in the input space, while in this paper we measure the data-point distribution from another perspective by using low dimensional slices to cut the input space and quantitatively analyzing the frequency of change in class labels on the cross sections. Our results also suggest that assumption made by I. J. Goodfellow et al. [2]: small perturbation to the values of many dimensions will accumulate and cause huge change to the output, might not be necessary for explaining the weakness since we only changed 1 pixel(i.e. values

1 pixel	0	1	2	3	4	5	6	7	8	9
0	0	18	27	33	13	19	10	13	32	32
1	11	0	3	10	1	8	7	5	12	26
2	24	3	0	39	14	37	26	11	6	6
3	36	13	40	0	35	88	39	23	25	36
4	38	6	43	61	0	54	29	35	26	31
5	11	1	32	86	22	0	31	30	12	14
6	21	12	39	56	27	36	0	6	25	20
7	10	6	5	23	15	41	9	0	6	20
8	54	34	16	23	8	14	21	5	0	49
9	25	19	5	18	11	16	7	14	24	0

3 pixels	0	1	2	3	4	5	6	7	8	9
0	0	31	43	47	19	31	25	20	52	43
1	19	0	7	12	4	11	9	8	16	31
2	41	11	0	52	24	44	38	25	20	20
3	65	31	58	0	55	108	67	45	47	54
4	60	12	59	71	0	68	57	48	40	49
5	24	11	49	104	37	0	46	50	19	30
6	33	25	51	66	40	53	0	10	34	38
7	22	9	12	35	25	48	13	0	14	29
8	77	50	20	36	17	23	31	14	0	64
9	37	32	14	27	16	21	14	22	33	0

5 pixels	0	1	2	3	4	5	6	7	8	9
0	0	36	48	54	24	37	32	29	54	52
1	28	0	7	14	5	12	12	13	21	47
2	51	16	0	56	33	49	43	35	22	26
3	71	45	69	0	63	119	77	58	56	61
4	73	21	64	76	0	73	64	55	44	57
5	47	17	59	108	52	0	57	60	26	46
6	43	33	60	76	52	62	0	20	37	45
7	25	12	15	36	28	55	15	0	13	36
8	84	63	24	39	19	29	33	18	0	77
9	45	43	19	30	22	27	19	23	39	0

Figure 6. Heat-maps that indicate the number of times of successful perturbation with the corresponding original-target class pair in 1, 3 and 5-pixel cases. Red index indicates the original classes and blue for target classes. The number from 0 to 9 indicates the class of airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck respectively. The cells with darker colors own higher numbers.

of few dimension) to successfully perturbed a considerable



Figure 7. The graphs count the total numbers of times of a specific class acting as the original(blue) and target(class for all successful perturbations, which the values are denoted by vertical axis. The horizontal axis indicates the index of each class which is the same to fig.6.

number of images.

We also noticed that the trained target classifier can give absolute probabilistic class label to a test image (i.e. assigning 1 probability to the target class and none for all other classes) where the gradient cannot be calculated. By using DE, such absoluteness can be easily broken by running a few generations of evolution. In addition, the success rate of perturbation might be further improved by having additional runs of our algorithm. Our algorithm and the naturally intriguing samples (i.e. very sensitive images) collected might be useful for generating better artificial adversarial samples for augmenting the training data set for the purpose of learning more robust models[26], which is left for future work.

## 6. Conclusion

In this paper we propose a differential evolution based method for generating adversarial images. Experimental results show that our proposal is effective on generating adversarial images in very limited conditions. For example, in an ideal case it can be done with just one-pixel perturbation out of 1024 pixels in total. We also discussed how our results can be beneficial for quantitatively understanding the geometrical features of DNN in high dimensional input.

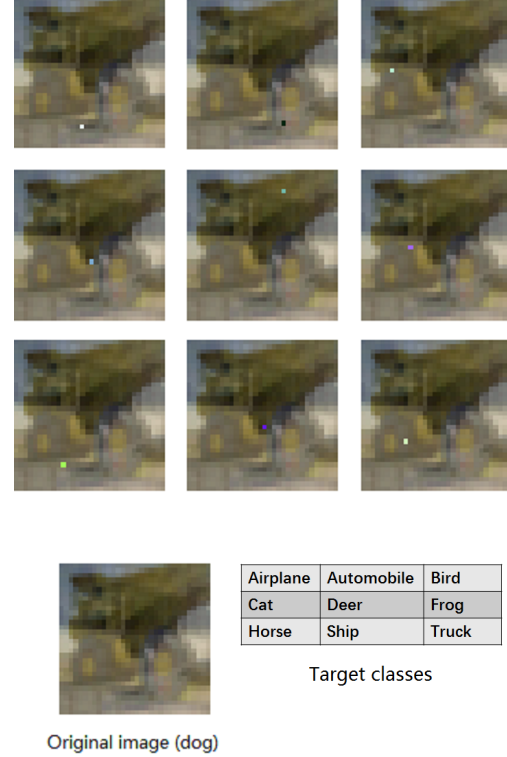


Figure 8. A confused image of dog that can be perturbed to all 9 target classes each with just one pixel modification by using our algorithm. The table in the bottom shows the class labels output by the target DNN, all with 100% confidence. This is curious and be very different to human since in our case the difference between these things is very clear. However, the perturbed pixels can be still noticed if looking carefully.

## 7. Acknowledgment

This research was partially supported by Collaboration Hubs for International Program (CHIRP) of SICORP, Japan Science and Technology Agency (JST).

## References

- [1] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B. and Swami, A., 2016, March. The limitations of deep learning in adversarial settings. In Security and Privacy, 2016 IEEE European Symposium on (pp. 372-387).
- [2] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
- [3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus. Intriguing properties of neural networks. In Proceedings of the 2014 Interna-



- tional Conference on Learning Representations. Computational and Biological Learning Society, 2014.
- [4] Moosavi-Dezfooli, S.M., Fawzi, A. and Frossard, P., 2016. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2574-2582).
- [5] Vargas, D.V. and Murata, J., 2016. Spectrum-Diverse Neuroevolution With Unified Neural Models. *IEEE transactions on neural networks and learning systems*.
- [6] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pages 1625. ACM, 2006.
- [7] Nguyen, A., Yosinski, J. and Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 427-436).
- [8] Bishop, C.M., 2006. Pattern recognition and machine learning. springer.
- [9] Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- [10] Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press. ISBN 9780585030944.
- [11] Krizhevsky, A., Nair, V. and Hinton, G., 2014. The CIFAR-10 dataset. online: <http://www.cs.toronto.edu/kriz/cifar.html>.
- [12] Storn, R. and Price, K., 1997. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), pp.341-359.
- [13] Narodytska, N. and Kasiviswanathan, S., 2017, July. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on (pp. 1310-1318). IEEE.
- [14] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar. The security of machine learning. *Machine Learning*, 81(2):121148, 2010. p.jpg
- [15] Zeiler, M. D and Fergus, R. Visualizing and understanding convolutional networks. *European conference on computer vision* (pp. 818-833).
- [16] Yosinski J, Clune J, Nguyen A, et al. Understanding neural networks through deep visualization[J]. *arXiv preprint arXiv:1506.06579*, 2015.
- [17] Wei D, Zhou B, Torralba A, et al. Understanding intra-class knowledge inside CNN[J]. *arXiv preprint arXiv:1507.02379*, 2015.
- [18] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps[J]. *arXiv preprint arXiv:1312.6034*, 2013.
- [19] Springenberg J T, Dosovitskiy A, Brox T, et al. Striving for simplicity: The all convolutional net[J]. *arXiv preprint arXiv:1412.6806*, 2014.
- [20] Papernot N, McDaniel P, Goodfellow I, et al. Practical black-box attacks against machine learning[C]//Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017: 506-519.
- [21] Dang H, Huang Y, Chang E C. Evading Classifiers by Morphing in the Dark[J]. 2017.
- [22] Fawzi, A., Moosavi-Dezfooli, S.M., Frossard, P. and Soatto, S., 2017. Classification regions of deep neural networks. *arXiv preprint arXiv:1705.09552*.
- [23] Fawzi, A., Moosavi-Dezfooli, S.M. and Frossard, P., 2017. A Geometric Perspective on the Robustness of Deep Networks (No. EPFL-ARTICLE-229872). Institute of Electrical and Electronics Engineers.
- [24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard, Universal adversarial perturbations, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto, Analysis of universal adversarial perturbations, *arXiv preprint arXiv:1705.09554*, 2017.
- [26] Andras Rozsa, Ethan M. Rudd, and Terrance E. Boult, Adversarial diversity and hard positive generation, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2016.